

# The Smart Sampling Kalman Filter with Symmetric Samples

Jannik Steinbring, Martin Pander, Uwe D. Hanebeck

*Intelligent Sensor-Actuator-Systems Laboratory (ISAS)  
Institute for Anthropomatics and Robotics  
Karlsruhe Institute of Technology (KIT), Germany*

---

## Abstract

Nonlinear Kalman Filters are powerful and widely-used techniques when trying to estimate the hidden state of a stochastic nonlinear dynamic system. In this paper, we extend the Smart Sampling Kalman Filter (S<sup>2</sup>KF) with a new point symmetric Gaussian sampling scheme. This not only improves the S<sup>2</sup>KF's estimation quality, but also reduces the time needed to compute the required optimal Gaussian samples drastically. Moreover, we improve the numerical stability of the sample computation, which allows us to accurately approximate a thousand-dimensional Gaussian distribution using tens of thousands of optimally placed samples. We evaluate the new symmetric S<sup>2</sup>KF by computing higher-order moments of standard normal distributions and investigate the estimation quality of the S<sup>2</sup>KF when dealing with symmetric measurement equations. Finally, extended object tracking based on many measurements per time step is considered. This high-dimensional estimation problem shows the advantage of the S<sup>2</sup>KF being able to use an arbitrary number of samples independent of the state dimension, in contrast to other state-of-the-art sample-based Kalman Filters.

---

## 1. Introduction

Estimating the hidden state of a stochastic dynamic system based on noisy measurements is crucial for many applications in control, object tracking, or robotics. When considering linear systems corrupted by additive Gaussian noise, the Kalman Filter (KF) is the optimal estimator with respect to the mean square error [1]. Unfortunately, most practical problems are nonlinear, making closed-form solutions intractable. Consequently, approximative approaches have to be used. Particle Filters (PFs) [2, 3, 4, 5] try to approximate the complete, in general multimodal, system state density with a set of weighted particles. This comes at the cost of computational complexity due to the curse of dimensionality. Another problem is sample degeneracy, in particular for high-dimensional state spaces, as a consequence of the particle reweighting using the likelihood function. To reduce computational complexity and circumvent the problem of sample degeneracy, the Progressive Gaussian Filter (PGF) [6] approximates the system state as a Gaussian and moves the particles automatically to the important regions of the state space. Nevertheless, those nonlinear filters are still costly compared to linear filters applied to nonlinear problems.

The Extended Kalman Filter (EKF) explicitly linearizes the underlying models around the current state estimate to be able to apply the standard KF to the considered problem [7]. Iterated variants of the EKF (IEKF) try to improve the EKF approach by iteratively searching for a more suitable point for the model linearization [7]. A more suitable way of model linearization is based on statistical linearization, which can be performed in the best case analytically or, in all other cases, by exploiting samples in the form of Linear Regression Kalman Filters (LRKFs) [8]. LRKFs obtain the required

---

*Email addresses:* jannik.steinbring@kit.edu (Jannik Steinbring), martin.pander@student.kit.edu (Martin Pander), uwe.hanebeck@ieee.org (Uwe D. Hanebeck)

moments by propagating samples through the system and measurement models and computing sample mean and sample covariance matrix, respectively. The most commonly used LRKF is the Unscented Kalman Filter (UKF) [9]. Its samples are, however, limited in number and placement, and several attempts exist to improve the UKF by finding its optimal parameter settings for specific estimation problems [10]. Nevertheless, the additional computational time required to find proper UKF parameters can be used instead to propagate more carefully chosen samples through the models in order to improve the estimation quality. For example, the Gauss-Hermite Kalman Filter (GHKF) introduced in [11] is based on the Gauss-Hermite quadrature rule to generate its samples. Unfortunately, the GHKF also suffers from the curse of dimensionality, and hence, is not well suited for larger state spaces. The fifth-degree Cubature Kalman Filter (CKF) [12] relies on a fifth-degree spherical-radial integration rule to construct its samples. However, by design, the number of samples still grows quadratically in the state dimension making the fifth-degree CKF computational burdensome when dealing with larger state spaces. A non-deterministic LRKF was proposed with the Randomized UKF (RUKF) [13, 14]. Here, an arbitrary number of randomly scaled and rotated UKF sample sets are combined to a single set of samples. On the one hand this has the advantage of being able to change the employed number of samples. On the other hand it prohibits a reproducible filter behavior and imposes an additional runtime overhead compared to other LRKFs due to the creation of several random orthogonal matrices per prediction and measurement update. The estimation quality of any LRKF, regardless of the sampling it is based on, can be improved by using the iterated statistical linearization approach [15]. A more detailed overview of linear filters and LRKFs can be found in [16, 17].

Recently, the Smart Sampling Kalman Filter ( $S^2KF$ ) was proposed in [18, 17], and already successfully used for Simultaneous Localization and Mapping (SLAM) in [19]. The  $S^2KF$  uses optimal deterministic sampling of a standard normal distribution comprising an arbitrary number of samples based on a combination of the Localized Cumulative Distribution (LCD) and a modified Cramér-von Mises distance [20, 21]. The same LCD approach was also extended to approximate arbitrary Gaussian mixture distributions [22]. In this paper, we extend the  $S^2KF$  with point symmetric Gaussian sampling. This new symmetric sampling reflects the point symmetry of the Gaussian distribution and allows for matching all odd moments of a standard normal distribution exactly, which results in a more accurate state estimation. We also improve the numerical stability of the LCD approach. As a result, it is now possible to compute an optimal approximation of a thousand-dimensional standard normal distribution comprising tens of thousands of samples. Moreover, due to the exploited point symmetry, the required number of samples that have to be optimized is reduced by half. Consequently, the new samples can be computed much faster. In this regard, the  $S^2KF$  catches up to state-of-the-art LRKFs as all of them also rely on a point symmetric sampling.

The remainder of the paper is organized as follows. First, we give a detailed problem formulation including the general recursive Bayesian estimation, its transition to Nonlinear Kalman Filters, and finally their approximation using LRKFs. After that, in Sec. 3, we introduce a new point symmetric version of the  $S^2KF$ . In Sec. 4, we evaluate the new symmetric  $S^2KF$  by computing higher-order moments of multivariate standard normal distributions, showing the advantage of the new point symmetric sampling scheme when dealing with symmetric measurement equations, and performing extended object tracking. Finally, conclusions are given in Sec. 5.

## 2. Problem Formulation

We consider estimating the hidden state  $\underline{x}_k$  of a discrete-time stochastic nonlinear dynamic system, where the system model

$$\underline{x}_k = \underline{a}_k(\underline{x}_{k-1}, \underline{w}_k) \quad (1)$$

describes its temporal evolution<sup>1</sup>. Additionally, we receive noisy measurements  $\tilde{y}_k$  that are assumed to be generated according to the measurement model

$$y_k = h_k(\underline{x}_k, v_k) . \quad (2)$$

Thus, the received measurements  $\tilde{y}_k$  are realizations of the random variable  $y_k$ . The noise variables  $w_k$  and  $v_k$  are assumed to be Gaussian and independent of the system state for all time steps.

Our goal is to determine a state estimate of  $\underline{x}_k$  in the form of a conditional state density

$$f_k^e(\underline{x}_k) := f(\underline{x}_k | \underline{y}_{k:1}) = f(\underline{x}_k | \tilde{y}_k, \dots, \tilde{y}_1)$$

recursively over time using Bayesian inference. Such a recursive estimator consists of two parts, namely the prediction step and the filter step. On the one hand, the prediction step propagates the state estimate  $f_{k-1}^e(\underline{x}_{k-1})$  from time step  $k-1$  to the current time step  $k$  by employing the system model (1) resulting in the predicted state density

$$f_k^p(\underline{x}_k) := f(\underline{x}_k | \underline{y}_{k-1:1}) = f(\underline{x}_k | \tilde{y}_{k-1}, \dots, \tilde{y}_1) .$$

On the other hand, the filter step incorporates a newly received measurement  $\tilde{y}_k$  into this propagated state estimate  $f_k^p(\underline{x}_k)$  with the aid of the measurement model (2).

To compute the prediction step, (1) has to be transformed into a state transition density according to

$$f_k^a(\underline{x}_k | \underline{x}_{k-1}) := \int \delta(\underline{x}_k - \underline{a}_k(\underline{x}_{k-1}, \underline{w}_k)) \cdot f_k^w(\underline{w}_k) d\underline{w}_k ,$$

where  $\delta(\cdot)$  denotes the Dirac delta distribution and  $f_k^w(\cdot)$  the system noise density

$$f_k^w(\underline{w}_k) = \mathcal{N}(\underline{w}_k; \hat{\underline{w}}_k, \mathbf{C}_k^w) .$$

Based on this, the predicted state density can be obtained with the Chapman-Kolmogorov equation [23]

$$\begin{aligned} f_k^p(\underline{x}_k) &= \int f_k^a(\underline{x}_k | \underline{x}_{k-1}) \cdot f_{k-1}^e(\underline{x}_{k-1}) d\underline{x}_{k-1} \\ &= \iint \delta(\underline{x}_k - \underline{a}_k(\underline{x}_{k-1}, \underline{w}_k)) \cdot f_{k-1}^e(\underline{x}_{k-1}) \cdot f_k^w(\underline{w}_k) d\underline{x}_{k-1} d\underline{w}_k . \end{aligned} \quad (3)$$

Concerning the filter step, Bayes' rule serves as its fundamental basis and the posterior, i.e., filtered, state density is obtained according to

$$f_k^e(\underline{x}_k) = \frac{f_k^h(\tilde{y}_k | \underline{x}_k) \cdot f_k^p(\underline{x}_k)}{f_k^y(\tilde{y}_k | \underline{y}_{k-1:1})} , \quad (4)$$

where  $f_k^h(\cdot)$  denotes the likelihood function and  $f_k^y(\cdot)$  the measurement distribution given all measurements up to the time step  $k-1$ . As  $f_k^y(\cdot)$  is independent of  $\underline{x}_k$ , it is only a normalization constant assuring that  $f_k^e(\underline{x}_k)$  is a valid density function. The required likelihood function is obtained in a similar manner to the state transition density according to

$$f_k^h(\tilde{y}_k | \underline{x}_k) := \int \delta(\tilde{y}_k - h_k(\underline{x}_k, v_k)) \cdot f_k^v(v_k) dv_k , \quad (5)$$

with the measurement noise density

$$f_k^v(v_k) = \mathcal{N}(v_k; \hat{v}_k, \mathbf{C}_k^v) .$$

---

<sup>1</sup>The subscript  $k$  denotes the discrete time step and vectors are underlined.

Given an initial state estimate  $f_0^e(\underline{x}_0)$ , the recursive state estimation of  $f_k^e(\underline{x}_k)$  is performed with the alternate use of the prediction step (3) and the filter step (4). However, computing both steps analytically is almost always impossible except for special cases such as linear models corrupted by additive Gaussian noise. Consequently, approximative solutions of the recursive Bayesian estimation have to be used instead.

### 2.1. Nonlinear Kalman Filters

Besides the popular Particle Filters, an important approximative Bayesian estimation technique is given by the class of Nonlinear Kalman Filters. These filters make two simplifications of the general Bayesian estimator described above.

First, the filtered as well as the predicted state estimates are always approximated as Gaussians. Hence, the predicted state density is given by

$$f_k^p(\underline{x}_k) \approx \mathcal{N}(\underline{x}_k; \hat{\underline{x}}_k^p, \mathbf{C}_k^p) ,$$

with predicted state mean

$$\begin{aligned} \hat{\underline{x}}_k^p &= \int \underline{x}_k \cdot f_k^p(\underline{x}_k) d\underline{x}_k \\ &= \iint \underline{a}_k(\underline{x}_{k-1}, \underline{w}_k) \cdot f_{k-1}^e(\underline{x}_{k-1}) \cdot f_k^w(\underline{w}_k) d\underline{x}_{k-1} d\underline{w}_k \end{aligned} \quad (6)$$

and predicted state covariance matrix

$$\begin{aligned} \mathbf{C}_k^p &= \int (\underline{x}_k - \hat{\underline{x}}_k^p) \cdot (\underline{x}_k - \hat{\underline{x}}_k^p)^\top \cdot f_k^p(\underline{x}_k) d\underline{x}_k \\ &= \iint (\underline{a}_k(\underline{x}_{k-1}, \underline{w}_k) - \hat{\underline{x}}_k^p) \cdot (\underline{a}_k(\underline{x}_{k-1}, \underline{w}_k) - \hat{\underline{x}}_k^p)^\top \cdot f_{k-1}^e(\underline{x}_{k-1}) \cdot f_k^w(\underline{w}_k) d\underline{x}_{k-1} d\underline{w}_k , \end{aligned} \quad (7)$$

respectively.

Second, the Bayesian filter step (4) can be reformulated in form of the joint density  $f_k^{x,y}(\underline{x}_k, \underline{y}_k)$  of predicted state  $\underline{x}_k$  and measurement  $\underline{y}_k$  according to

$$f_k^e(\underline{x}_k) = \frac{f_k^{x,y}(\underline{x}_k, \underline{y}_k | \underline{\tilde{y}}_{k-1:1})}{f_k^y(\underline{\tilde{y}}_k | \underline{\tilde{y}}_{k-1:1})} .$$

In Kalman filtering, this joint density is also approximated as a Gaussian resulting in the approximative Gaussian posterior state density

$$f_k^e(\underline{x}_k) \approx \frac{\mathcal{N}\left(\begin{bmatrix} \underline{x}_k \\ \underline{\tilde{y}}_k \end{bmatrix}; \begin{bmatrix} \hat{\underline{x}}_k^p \\ \hat{\underline{y}}_k \end{bmatrix}, \begin{bmatrix} \mathbf{C}_k^p & \mathbf{C}_k^{x,y} \\ (\mathbf{C}_k^{x,y})^\top & \mathbf{C}_k^y \end{bmatrix}\right)}{f_k^y(\underline{\tilde{y}}_k | \underline{\tilde{y}}_{k-1:1})} = \mathcal{N}(\underline{x}_k; \hat{\underline{x}}_k^e, \mathbf{C}_k^e) ,$$

with posterior state mean

$$\hat{\underline{x}}_k^e = \hat{\underline{x}}_k^p + \mathbf{C}_k^{x,y} \cdot (\mathbf{C}_k^y)^{-1} \cdot (\underline{\tilde{y}}_k - \hat{\underline{y}}_k) \quad (8)$$

and posterior state covariance matrix

$$\mathbf{C}_k^e = \mathbf{C}_k^p - \mathbf{C}_k^{x,y} \cdot (\mathbf{C}_k^y)^{-1} \cdot (\mathbf{C}_k^{x,y})^\top , \quad (9)$$

which are the well-known Kalman Filter formulas[7]. In order to obtain (8) and (9), the measurement mean

$$\begin{aligned}\hat{y}_k &= \int \underline{y}_k \cdot f_k^y(\underline{y}_k) \, d\underline{y}_k \\ &= \iint \underline{h}_k(\underline{x}_k, \underline{v}_k) \cdot f_k^p(\underline{x}_k) \cdot f_k^v(\underline{v}_k) \, d\underline{x}_k \, d\underline{v}_k \, ,\end{aligned}\tag{10}$$

the measurement covariance matrix

$$\begin{aligned}\mathbf{C}_k^y &= \int (\underline{y}_k - \hat{y}_k) \cdot (\underline{y}_k - \hat{y}_k)^\top \cdot f_k^y(\underline{y}_k) \, d\underline{y}_k \\ &= \iint (\underline{h}_k(\underline{x}_k, \underline{v}_k) - \hat{y}_k) \cdot (\underline{h}_k(\underline{x}_k, \underline{v}_k) - \hat{y}_k)^\top \cdot f_k^p(\underline{x}_k) \cdot f_k^v(\underline{v}_k) \, d\underline{x}_k \, d\underline{v}_k \, ,\end{aligned}\tag{11}$$

as well as the cross-covariance matrix of predicted state and measurement

$$\begin{aligned}\mathbf{C}_k^{x,y} &= \iint (\underline{x}_k - \hat{x}_k^p) \cdot (\underline{y}_k - \hat{y}_k)^\top \cdot f_k^{x,y}(\underline{x}_k, \underline{y}_k) \, d\underline{x}_k \, d\underline{y}_k \\ &= \iint (\underline{x}_k - \hat{x}_k^p) \cdot (\underline{h}_k(\underline{x}_k, \underline{v}_k) - \hat{y}_k)^\top \cdot f_k^p(\underline{x}_k) \cdot f_k^v(\underline{v}_k) \, d\underline{x}_k \, d\underline{v}_k\end{aligned}\tag{12}$$

are required.

In summary, running a Nonlinear Kalman Filter boils down to computing two integrals for the prediction step and three integrals for the filter step. Moreover, note that no explicit likelihood function (5) is required at all, which eases filter development.

## 2.2. Linear Regression Kalman Filters

Unfortunately, computing the above integrals in closed-form is only possible for a small set of system and measurement models. This includes for example polynomials, trigonometric and, of course, linear functions (leading to the classical linear Kalman Filter). The closed-form integration yields the best possible Kalman Filter for the given models. In all other cases, numerical integration methods have to be applied, which can result in a reduced estimation performance and an increased computational demand.

As we aim for an online estimation technique, the employed integration method has to possess a *real-time capable computational complexity* and still deliver adequate integration results in order to obtain a good recursive state estimation quality. When looking at the five integrals in Sec. 2.1, it can be seen that the last terms are always a product of two independent Gaussian densities, namely

$$f_{k-1}^e(\underline{x}_{k-1}) \cdot f_k^w(\underline{w}_k) = \mathcal{N} \left( \begin{bmatrix} \underline{x}_{k-1} \\ \underline{w}_k \end{bmatrix}; \begin{bmatrix} \hat{\underline{x}}_{k-1}^e \\ \hat{\underline{w}}_k \end{bmatrix}, \begin{bmatrix} \mathbf{C}_{k-1}^e & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_k^w \end{bmatrix} \right)\tag{13}$$

for the prediction and

$$f_k^p(\underline{x}_k) \cdot f_k^v(\underline{v}_k) = \mathcal{N} \left( \begin{bmatrix} \underline{x}_k \\ \underline{v}_k \end{bmatrix}; \begin{bmatrix} \hat{\underline{x}}_k^p \\ \hat{\underline{v}}_k \end{bmatrix}, \begin{bmatrix} \mathbf{C}_k^p & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_k^v \end{bmatrix} \right)\tag{14}$$

for the measurement update, respectively.

By exploiting this fact, an efficient, i.e., fast but still accurate, computation of the integrals is possible. This can be done by replacing the occurring Gaussian distributions (13) and (14) with proper Dirac mixture densities, that is, sample-based density representations, and evaluating the system model (1) and measurement model (2) using these samples. As a result, emphasis is directly put on the important regions of the state space, and the regions covered by only a small portion of the probability mass of the Gaussian densities are neglected. This approach leads to the class of Linear Regression Kalman Filters (LRKFs).

A Dirac mixture approximation of a given probability density function  $f_k(\underline{s}_k)$  comprising  $M_k$  samples with sample positions  $\underline{s}_{k,i}$  and sample weights  $\alpha_{k,i}$  is defined as

$$\sum_{i=1}^{M_k} \alpha_{k,i} \cdot \delta(\underline{s}_k - \underline{s}_{k,i}) , \quad (15)$$

where for the sample weights

$$\sum_{i=1}^{M_k} \alpha_{k,i} = 1$$

holds. Such an approximation can be computed in several ways, e.g., by simply using random sampling or deterministic approaches such as done by the UKF.

Now, we assume that an approximation

$$\sum_{i=1}^{M_k} \alpha_{k,i} \cdot \delta \left( \begin{bmatrix} \underline{x}_{k-1} \\ \underline{w}_k \end{bmatrix} - \begin{bmatrix} \underline{x}_{k-1,i} \\ \underline{w}_{k,i} \end{bmatrix} \right) \quad (16)$$

of the Gaussian joint density (13) comprising  $M_k$  samples with positions  $[\underline{x}_{k-1,i}^\top, \underline{w}_{k,i}^\top]^\top$  and weights  $\alpha_{k,i}$  is at hand. By replacing the Gaussian joint density in the integrals (6) and (7) with the Dirac mixture approximation (16), and using the Dirac sifting property, we obtain an approximation for the predicted state mean

$$\hat{\underline{x}}_k^p \approx \sum_{i=1}^{M_k} \alpha_{k,i} \cdot \underline{a}_k(\underline{x}_{k-1,i}, \underline{w}_{k,i})$$

and the predicted state covariance matrix

$$\mathbf{C}_k^p \approx \sum_{i=1}^{M_k} \alpha_{k,i} \cdot (\underline{a}_k(\underline{x}_{k-1,i}, \underline{w}_{k,i}) - \hat{\underline{x}}_k^p) \cdot (\underline{a}_k(\underline{x}_{k-1,i}, \underline{w}_{k,i}) - \hat{\underline{x}}_k^p)^\top .$$

The same procedure is used for computing the integrals required for the measurement update. First, a Dirac mixture approximation

$$\sum_{i=1}^{M_k} \alpha_{k,i} \cdot \delta \left( \begin{bmatrix} \underline{x}_k \\ \underline{v}_k \end{bmatrix} - \begin{bmatrix} \underline{x}_{k,i} \\ \underline{v}_{k,i} \end{bmatrix} \right) , \quad (17)$$

of the Gaussian (14) encompassing  $M_k$  samples with positions  $[\underline{x}_{k,i}^\top, \underline{v}_{k,i}^\top]^\top$  and weights  $\alpha_{k,i}$  is computed. Second, by replacing the joint Gaussian with its Dirac mixture approximation in the three integrals (10), (11), and (12), and using once more the Dirac sifting property, we get an approximation for the measurement mean

$$\hat{\underline{y}}_k \approx \sum_{i=1}^{M_k} \alpha_{k,i} \cdot \underline{h}_k(\underline{x}_{k,i}, \underline{v}_{k,i}) ,$$

the measurement covariance matrix

$$\mathbf{C}_k^y \approx \sum_{i=1}^{M_k} \alpha_{k,i} \cdot (\underline{h}_k(\underline{x}_{k,i}, \underline{v}_{k,i}) - \hat{\underline{y}}_k) \cdot (\underline{h}_k(\underline{x}_{k,i}, \underline{v}_{k,i}) - \hat{\underline{y}}_k)^\top ,$$

and the cross-covariance matrix

$$\mathbf{C}_k^{x,y} \approx \sum_{i=1}^{M_k} \alpha_{k,i} \cdot (\underline{x}_{k,i} - \hat{\underline{x}}_k^p) \cdot (\underline{h}_k(\underline{x}_{k,i}, \underline{v}_{k,i}) - \hat{\underline{y}}_k)^\top .$$

It should be noted that the number of samples for the time and the measurement update do not have to be the same. Moreover, the Dirac mixture approximations (16) and (17) can be completely different in the way they are obtained, although this is usually not the case.

### 3. The Smart Sampling Kalman Filter with Symmetric Samples

In [20], the authors proposed an approach based on the Localized Cumulative Distribution (LCD) to optimally approximate Gaussian distributions with a set of equally weighted samples. This is done by transforming the approximation problem into an optimization problem. Unfortunately, such optimization is very time-consuming, and hence, not suitable for online nonlinear filtering. To enable the LCD approach for online filtering, it is used to optimally sample only a standard normal distribution offline (before filter usage) and transform these samples online (during filter usage) to any required Gaussian with the aid of the Mahalanobis transformation [24]. This is the fundamental basis for the S<sup>2</sup>KF [17]. But other nonlinear estimators such as the Progressive Gaussian Filter also make use of this sampling technique.

However, the current LCD approach can, and will, arrange the samples in an arbitrary way to optimally approximate a standard normal distribution. More precisely, it does not take the point symmetry of the standard normal distribution explicitly into account. This can lead to a set of asymmetric samples. Here, we extend the LCD approach to approximate an  $N$ -dimensional standard normal distribution with a set of *point symmetric* and equally weighted samples. Moreover, we improve the numerical stability of the LCD approach to allow approximations of very high dimensions. This new optimal point symmetric sampling is then used to obtain a symmetric version of the S<sup>2</sup>KF.

The use of point symmetric samples offers several benefits. First, the symmetric samples reflect the symmetry of the standard normal distribution allowing for more accurate estimation results as will be seen in the evaluation. Second, the used point symmetry makes it possible to capture *all odd moments* of the standard normal distribution exactly (a proof is given in A). Finally, the required number of sample positions that have to be optimized is reduced by half. Consequently, an approximation can be computed much faster. Besides point symmetry, other symmetries such as axial symmetry could also be exploited. However, this would prevent us from using an arbitrary number of samples and would limit the optimizer's control over the sample placement.

In the following, we first define the set of parameters describing point symmetric Dirac mixtures in Sec. 3.1. These parameters have then to be optimized in order to approximate a standard normal distribution in an optimal way. This requires distance measures between a standard normal distribution and the point symmetric Dirac mixtures given in Sec. 3.2. Subsequently, the gradients of the distance measures are derived in Sec. 3.3. Finally, in Sec. 3.4, we give a procedure to compute point symmetric Dirac mixture approximations of standard normal distributions based on the introduced distance measures and their gradients.

#### 3.1. Point Symmetric Dirac Mixtures

First, we have to modify the generic Dirac mixture (15) to obtain a symmetric one. This is performed by distinguishing between an even and odd number of samples. For the case of  $2L$  samples with  $L \in \mathbb{N}_+$ , that is, the even case, we place the samples point symmetrically around the state space origin yielding the equally weighted Dirac mixture

$$\frac{1}{2L} \sum_{i=1}^L \delta(\underline{s} - \underline{s}_i) + \delta(\underline{s} + \underline{s}_i) ,$$

with sample positions  $\underline{s}_i$  and  $-\underline{s}_i$ . For  $2L + 1$  samples, the odd case, we additionally place a sample fixed at the state space origin and obtain the Dirac mixture

$$\frac{1}{2L + 1} \left( \delta(\underline{s}) + \sum_{i=1}^L \delta(\underline{s} - \underline{s}_i) + \delta(\underline{s} + \underline{s}_i) \right) .$$

This preserves the desired point symmetry. As the position of the additional sample in the odd case is constant, the set of parameters

$$S := \{\underline{s}_1, \dots, \underline{s}_L\}$$

is the same for both Dirac mixtures.

Note that the UKF sample set (with equally weighted samples) is a special case of these point symmetric Dirac mixtures [9]. With an even number of samples, it has the parametrization

$$\underline{s}_i = \sqrt{N} \cdot \underline{e}_i \quad \forall i \in \{1, \dots, N\} ,$$

where  $\underline{e}_i$  denotes the unit vector along the  $i$ -th dimension. In the odd case, the parametrization is

$$\underline{s}_i = \sqrt{N + 0.5} \cdot \underline{e}_i \quad \forall i \in \{1, \dots, N\} ,$$

that is, the sample spread is larger due to the additional point mass at the state space origin.

### 3.2. Distance Measures

Our goal is to determine the set of parameters  $S$  for the above Dirac mixtures so that they approximate a multivariate standard normal distribution in an optimal way. This requires a distance measure between the involved continuous and discrete distributions. As the classical cumulative distribution function is not suitable for the multi-dimensional case [25], we utilize the LCD approach in the same way as the asymmetric S<sup>2</sup>KF.

**Definition 3.1** (Localized Cumulative Distribution [17]).

Let  $f(\underline{s})$  be a  $N$ -dimensional density function. The corresponding Localized Cumulative Distribution is defined as

$$F(\underline{m}, b) = \int_{\mathbb{R}^N} f(\underline{s}) \cdot K(\underline{s} - \underline{m}, b) \, d\underline{s} ,$$

with  $\underline{m} \in \mathbb{R}^N$ ,  $b \in \mathbb{R}_+$ , and the symmetric and integrable kernel

$$K(\underline{s} - \underline{m}, b) = \exp \left( -\frac{1}{2} \frac{\|\underline{s} - \underline{m}\|_2^2}{b^2} \right) .$$

Here,  $\underline{m}$  characterizes the location of the kernel and  $b$  its size.

The LCD of an  $N$ -dimensional standard normal distribution is an integral of a product of two (unnormalized) Gaussians. By using the fact that the product of two Gaussian distributions is also an unnormalized Gaussian and the integral over a probability density equals one, its LCD is obtained by [20]

$$\begin{aligned} F_N(\underline{m}, b) &= \int_{\mathbb{R}^N} \mathcal{N}(\underline{s}; \underline{0}, \mathbf{I}_N) \cdot (2\pi)^{\frac{N}{2}} b^N \mathcal{N}(\underline{s}; \underline{m}, b^2 \mathbf{I}_N) \, d\underline{s} \\ &= \frac{(2\pi)^{\frac{N}{2}} b^N}{(2\pi)^{\frac{N}{2}} \sqrt{|(1+b^2)\mathbf{I}_N|}} \exp \left( -\frac{1}{2} \frac{\|\underline{m}\|_2^2}{(1+b^2)} \right) \\ &= \left( \frac{b^2}{1+b^2} \right)^{\frac{N}{2}} \exp \left( -\frac{1}{2} \frac{\|\underline{m}\|_2^2}{(1+b^2)} \right) , \end{aligned}$$

where  $\mathbf{I}_N$  denotes the identity matrix of dimension  $N$ . Based on the Dirac sifting property, the LCD of the Dirac mixture comprising an even number of samples is given by

$$F_\delta^e(S, \underline{m}, b) = \frac{1}{2L} \left( \sum_{i=1}^L \exp \left( -\frac{1}{2} \frac{\|\underline{s}_i - \underline{m}\|_2^2}{b^2} \right) + \exp \left( -\frac{1}{2} \frac{\|\underline{0} - \underline{m}\|_2^2}{b^2} \right) \right) ,$$



whereas the LCD of the odd Dirac mixture is

$$F_\delta^o(S, \underline{m}, b) = \frac{1}{2L+1} \left( \exp \left( -\frac{1}{2} \frac{\|\underline{m}\|_2^2}{b^2} \right) + \sum_{i=1}^L \exp \left( -\frac{1}{2} \frac{\|\underline{s}_i - \underline{m}\|_2^2}{b^2} \right) + \exp \left( -\frac{1}{2} \frac{\|-\underline{s}_i - \underline{m}\|_2^2}{b^2} \right) \right).$$

To compare the standard normal LCD with a Dirac mixture LCD, we use the modified Cramér-von Mises distance defined as follows.

**Definition 3.2** (Modified Cramér-von Mises Distance).

The modified Cramér-von Mises (CvM) distance  $D$  between two LCDs  $F(\underline{m}, b)$  and  $\tilde{F}(\underline{m}, b)$  is given by

$$D(F, \tilde{F}) = \int_0^\infty w(b) \int_{\mathbb{R}^N} \left( F(\underline{m}, b) - \tilde{F}(\underline{m}, b) \right)^2 d\underline{m} db ,$$

with weighting function

$$w(b) = \begin{cases} \pi^{-\frac{N}{2}} b^{1-N} & , b \in [0, b_{\max}] \\ 0 & , \text{elsewhere} . \end{cases}$$

The new term  $\pi^{-\frac{N}{2}}$  in the weighting function  $w(b)$  (in contrast with the definition in [17]) is a consequence of the involved LCDs  $F_{\mathcal{N}}$ ,  $F_\delta^e$ , and  $F_\delta^o$ . Without this term, the modified CvM distances between these LCDs would be unbounded for an increasing dimension  $N$ , which in turn would make the distances numerically unstable. This improvement now allows the S<sup>2</sup>KF to compute Dirac mixture approximations for very high state dimensions, e.g.,  $N > 200$ .

First, we consider the distance between the standard normal distribution and the Dirac mixture comprising an even number of samples, and then extend the results to the odd case. The distance  $D(F_{\mathcal{N}}, F_\delta^e)$  can be split into three terms according to

$$D(F_{\mathcal{N}}, F_\delta^e) = D^e(S) = D_1^e - 2D_2^e(S) + D_3^e(S) ,$$

with the sample-independent part

$$D_1^e = \int_0^{b_{\max}} b \left( \frac{b^2}{1+b^2} \right)^{\frac{N}{2}} db ,$$

and the sample-dependent terms

$$D_2^e(S) = \int_0^{b_{\max}} \frac{2b}{2L} \left( \frac{2b^2}{1+2b^2} \right)^{\frac{N}{2}} \cdot \sum_{i=1}^L \exp \left( -\frac{1}{2} \frac{\|\underline{s}_i\|_2^2}{(1+2b^2)} \right) db ,$$

and

$$D_3^e(S) = \int_0^{b_{\max}} \frac{2b}{(2L)^2} \sum_{i=1}^L \sum_{j=1}^L \exp \left( -\frac{1}{2} \frac{\|\underline{s}_i - \underline{s}_j\|_2^2}{2b^2} \right) + \exp \left( -\frac{1}{2} \frac{\|\underline{s}_i + \underline{s}_j\|_2^2}{2b^2} \right) db .$$

The proof is given in B. Note that the integration over  $b$  is bounded by  $b_{\max}$  due to the support of the weighting function  $w(b)$ . To speed up the distance computation, the following theorem can be applied.

**Theorem 3.1.**

For a given  $b_{\max}$ , the following expression for  $D_3^e(S)$  can be obtained

$$D_3^e(S) = \frac{2}{(2L)^2} \sum_{i=1}^L \sum_{j=1}^L \frac{b_{\max}^2}{2} \left( \exp \left( -\frac{1}{2} \frac{\|\underline{s}_i - \underline{s}_j\|_2^2}{2b_{\max}^2} \right) + \exp \left( -\frac{1}{2} \frac{\|\underline{s}_i + \underline{s}_j\|_2^2}{2b_{\max}^2} \right) \right) + \frac{1}{8} \left( \|\underline{s}_i - \underline{s}_j\|_2^2 \text{Ei} \left( -\frac{1}{2} \frac{\|\underline{s}_i - \underline{s}_j\|_2^2}{2b_{\max}^2} \right) + \|\underline{s}_i + \underline{s}_j\|_2^2 \text{Ei} \left( -\frac{1}{2} \frac{\|\underline{s}_i + \underline{s}_j\|_2^2}{2b_{\max}^2} \right) \right) ,$$

where  $\text{Ei}(\cdot)$  denotes the exponential integral.

*Proof.* The proof is given in C. □

Now, we consider the case of an odd number of samples. Like in the even case,  $D(F_{\mathcal{N}}, F_{\delta}^o)$  can be split into three terms

$$D(F_{\mathcal{N}}, F_{\delta}^o) = D^o(S) = D_1^o - 2D_2^o(S) + D_3^o(S) .$$

The first part  $D_1^o$  is also independent of the samples  $S$  and identical to its even counterpart, i.e.,

$$D_1^o = D_1^e .$$

The sample-dependent terms  $D_2^o(S)$  and  $D_3^o(S)$  can be expressed in terms of the even case plus additional terms due to the fixed sample at the state space origin according to

$$D_2^o(S) = \frac{2L}{2L+1} D_2^e(S) + \int_0^{b_{\max}} \frac{b}{2L+1} \left( \frac{2b^2}{1+2b^2} \right)^{\frac{N}{2}} db$$

and

$$D_3^o(S) = \frac{(2L)^2}{(2L+1)^2} D_3^e(S) + \frac{b_{\max}^2}{2(2L+1)^2} + \int_0^{b_{\max}} \frac{4b}{(2L+1)^2} \sum_{i=1}^L \exp \left( -\frac{1}{2} \frac{\|\underline{s}_i\|_2^2}{2b^2} \right) db .$$

The proof is given in D. Like for the even case, also the computation of the odd case can be sped up by using the following theorem.

**Theorem 3.2.**

For a given  $b_{\max}$ , the following expression for  $D_3^o(S)$  can be obtained

$$D_3^o(S) = \frac{(2L)^2}{(2L+1)^2} D_3^e(S) + \frac{b_{\max}^2}{2(2L+1)^2} + \frac{4}{(2L+1)^2} \sum_{i=1}^L \frac{b_{\max}^2}{2} \exp \left( -\frac{1}{2} \frac{\|\underline{s}_i\|_2^2}{2b_{\max}^2} \right) + \frac{1}{8} \|\underline{s}_i\|_2^2 \text{Ei} \left( -\frac{1}{2} \frac{\|\underline{s}_i\|_2^2}{2b_{\max}^2} \right) ,$$

where  $\text{Ei}(\cdot)$  denotes the exponential integral.

*Proof.* The proof is given in E. □

The extra terms in  $D_2^o(S)$  and  $D_3^o(S)$ , compared to the even case, reflect the influence of the additional sample, placed at the state space origin, on the distance between the Dirac mixture and the standard normal distribution. The result is that the point mass of the additional sample will cause the other samples to have a slightly larger spread compared to a sample set without the additional sample at the state space origin. Concerning the above mentioned numerical stability, we also give a proof for the boundedness of both distances  $D^e(S)$  and  $D^o(S)$  in F.

### 3.3. Gradients of the Distance Measures

In order to optimize the parameters  $S$  of a given Dirac mixture, we chose to apply a gradient-based iterative optimization procedure. This requires the partial derivatives of the two distance measures  $D^e(S)$  and  $D^o(S)$  with respect to the set of parameters  $S$ . For the even case, the partial derivatives are

$$\frac{\partial D^e(S)}{\partial s_i^{(d)}} = -2 \frac{\partial D_2^e(S)}{\partial s_i^{(d)}} + \frac{\partial D_3^e(S)}{\partial s_i^{(d)}} \quad \forall d \in \{1, \dots, N\} ,$$

with its two terms

$$\frac{\partial D_2^e(S)}{\partial s_i^{(d)}} = -\frac{s_i^{(d)}}{2L} \int_0^{b_{\max}} \frac{2b}{(1+2b^2)} \left( \frac{2b^2}{1+2b^2} \right)^{\frac{N}{2}} \cdot \exp \left( -\frac{1}{2} \frac{\|\underline{s}_i\|_2^2}{(1+2b^2)} \right) db ,$$

and

$$\frac{\partial D_3^e(S)}{\partial s_i^{(d)}} = -\frac{2}{(2L)^2} \int_0^{b_{\max}} \frac{1}{b} \cdot \sum_{j=1}^L (s_i^{(d)} - s_j^{(d)}) \exp\left(-\frac{1}{2} \frac{\|\underline{s}_i - \underline{s}_j\|_2^2}{2b^2}\right) + (s_i^{(d)} + s_j^{(d)}) \exp\left(-\frac{1}{2} \frac{\|\underline{s}_i + \underline{s}_j\|_2^2}{2b^2}\right) db .$$

Analogous to  $D_3^e(S)$ , the following theorem can be used for the computation of its partial derivatives.

**Theorem 3.3.**

For a given  $b_{\max}$ , the following expression for  $\frac{\partial D_3^e(S)}{\partial s_i^{(d)}}$  can be obtained

$$\frac{\partial D_3^e(S)}{\partial s_i^{(d)}} = \frac{1}{(2L)^2} \sum_{j=1}^L (s_i^{(d)} - s_j^{(d)}) \text{Ei}\left(-\frac{1}{2} \frac{\|\underline{s}_i - \underline{s}_j\|_2^2}{2b_{\max}^2}\right) + (s_i^{(d)} + s_j^{(d)}) \text{Ei}\left(-\frac{1}{2} \frac{\|\underline{s}_i + \underline{s}_j\|_2^2}{2b_{\max}^2}\right) ,$$

where  $\text{Ei}(\cdot)$  denotes the exponential integral.

*Proof.* The proof is given in G. □

As with the distance  $D^o(S)$  itself, its partial derivatives

$$\frac{\partial D^o(S)}{\partial s_i^{(d)}} = -2 \frac{\partial D_2^o(S)}{\partial s_i^{(d)}} + \frac{\partial D_3^o(S)}{\partial s_i^{(d)}} \quad \forall d \in \{1, \dots, N\}$$

can be obtained in terms of the even case plus additional terms according to

$$\frac{\partial D_2^o(S)}{\partial s_i^{(d)}} = \frac{2L}{2L+1} \frac{\partial D_2^e(S)}{\partial s_i^{(d)}}$$

and

$$\frac{\partial D_3^o(S)}{\partial s_i^{(d)}} = \frac{(2L)^2}{(2L+1)^2} \frac{\partial D_3^e(S)}{\partial s_i^{(d)}} - \frac{2s_i^{(d)}}{(2L+1)^2} \int_0^{b_{\max}} \frac{1}{b} \exp\left(-\frac{1}{2} \frac{\|\underline{s}_i\|_2^2}{2b^2}\right) db .$$

To ease the computation of the partial derivatives of  $D_3^o(S)$ , the next theorem can be used.

**Theorem 3.4.**

For a given  $b_{\max}$ , the following expression for  $\frac{\partial D_3^o(S)}{\partial s_i^{(d)}}$  can be obtained

$$\frac{\partial D_3^o(S)}{\partial s_i^{(d)}} = \frac{(2L)^2}{(2L+1)^2} \frac{\partial D_3^e(S)}{\partial s_i^{(d)}} + \frac{s_i^{(d)}}{(2L+1)^2} \text{Ei}\left(-\frac{1}{2} \frac{\|\underline{s}_i\|_2^2}{2b_{\max}^2}\right) ,$$

where  $\text{Ei}(\cdot)$  denotes the exponential integral.

*Proof.* The proof is given in H. □

### 3.4. The $S^2KF$ with Symmetric Samples

After defining the distance measures  $D^e(S)$  and  $D^o(S)$ , including their partial derivatives, we can compute a Dirac mixture approximation of a standard normal distribution comprising an arbitrary number of optimally placed point symmetric samples.

To achieve this, we utilize the low memory BFGS quasi-Newton optimization (L-BFGS) [26]. The low memory variant is essential here, as it avoids the explicit computation and storage of the Hessian matrix. The set of Dirac mixture parameters  $S$  encompasses  $L \times N$  single parameters to be optimized. Hence, the Hessian matrix of  $D^e(S)$  or  $D^o(S)$  would contain  $(L \times N)^2$  entries. When now assuming only a linear increase in the number of samples for an increasing dimension  $N$ , that is,  $2L = C \cdot N$ ,

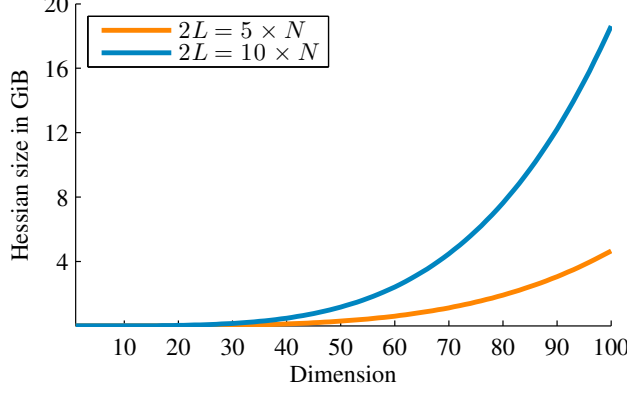


Figure 1: Size of the Hessian matrix for different dimensions and number of samples.

with a linear factor  $C$ , the size of the Hessian grows with  $\mathcal{O}(N^4)$ . This problem is illustrated in Fig. 1 for two different linear factors (5 and 10). It can be seen that approximating a 100-dimensional standard normal distribution with a thousand samples would require a Hessian of  $\approx 20$  gigabytes, and already a Hessian of over 4 gigabytes in case of only 500 samples. Consequently, using the Hessian directly in the optimization is intractable.

The computation of the point symmetric samples works as follows.

1. Choose the desired number of samples  $M$  to approximate the  $N$ -dimensional standard normal distribution.
2. Depending on the number of samples, the even or odd distance measure is selected.
3. A proper maximum kernel width  $b_{\max}$  has to be selected. Generally speaking, the larger the dimension  $N$  is the larger  $b_{\max}$  has to be in order to consider all sample positions during the optimization, and thus, to get a meaningful approximation. Empirically, we have found that a value of 70 is large enough for up to  $N \leq 1000$  dimensions.
4. An initial set of Dirac mixture parameters is obtained by drawing  $L$  samples randomly from an  $N$ -dimensional standard normal distribution.
5. The L-BFGS procedure optimizes the Dirac mixture parameters such that the distance measure is minimized, i.e., it moves the initial samples in the state space to approximate the standard normal distribution in an optimal way. The Dirac mixture parameters resulting from the L-BFGS procedure are denoted as  $\{z_i\}_{i=1}^L$ .
6. The Dirac mixture approximation given by the set of parameters  $\{z_i\}_{i=1}^L$  finally undergoes a Mahalanobis transformation so that the transformed Dirac mixture captures the identity covariance matrix of the standard normal distribution as much as possible. This is necessary, as the proposed distance measures do not explicitly consider the covariance matrix as a constraint. The transformation is done by computing the sample covariance matrix

$$\mathbf{C}^z = \frac{2}{M} \sum_{i=1}^L z_i \cdot z_i^\top ,$$

and scaling each sample according to

$$s_i = \sqrt{(\mathbf{C}^z)^{-1}} \cdot z_i \quad \forall i \in \{1, \dots, L\} ,$$

where  $\sqrt{(\mathbf{C}^z)^{-1}}$  denotes the Cholesky decomposition of the inverse sample covariance matrix. The Dirac mixture defined by the parameters  $\{s_i\}_{i=1}^L$  is the final approximation of the standard normal distribution.

Experimentally, we have found that in situations where the covariance matrix was added as an explicit constraint to the optimization procedure, the sample covariance matrix of the resulting Dirac mixture

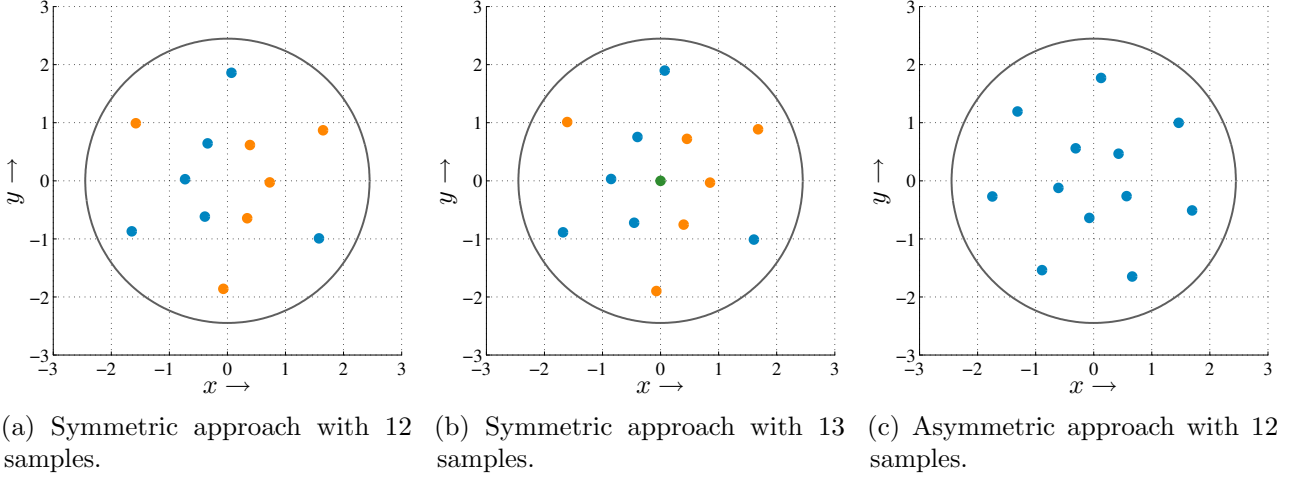


Figure 2: Different LCD-based approximations of a two-dimensional standard normal distribution with samples  $\underline{s}_i$  (blue), point symmetric counterparts  $-\underline{s}_i$  (orange), fixed sample at the state space origin in the odd case (green), and 95% confidence interval of the standard normal distribution (gray).

was less accurate compared to the proposed Mahalanobis approach. Moreover, the constraint made the optimization procedure much more time-consuming. Consequently, we dropped this approach in favor of the Mahalanobis transformation.

The results of different LCD-based approximations of a two-dimensional standard normal distribution are depicted in Fig. 2. On the one hand, Figures 2a and 2b show approximations using the new symmetric sampling scheme comprising 12 and 13 samples, respectively. The point symmetric arrangement around the state space origin can be clearly seen. Note also the subtle difference in the sample spread of the samples near the state space origin between Fig. 2a and Fig. 2b. This is caused by the additional point mass from the fixed sample at the state space origin. On the other hand, Fig. 2c shows an approximation based on the classical asymmetric sampling scheme also comprising 12 samples. Here, the optimization procedure can position all samples individually, and hence, the samples are not necessarily arranged in a point symmetric way like in the depicted case.

Using the above described optimal point symmetric sampling of a standard normal distribution, we obtain a symmetric version of the  $S^2KF$ . Furthermore, to avoid a re-computation on every program start, we store any computed Dirac mixture approximation of a standard normal distribution persistent in the file system for later reuse. This mechanism is called the *Sample Cache* and was already used by the asymmetric  $S^2KF$ .

## 4. Evaluation

In this Section, we want to compare the new point symmetric sampling scheme of the  $S^2KF$  with its asymmetric version and other state-of-the-art LRKFs. First, we take a closer look at the approximation of higher-order moments of standard normal distributions. Then, the advantage of using a point symmetric sampling scheme, and hence, the new version of the  $S^2KF$ , is discussed by means of a simple symmetric measurement equation. Finally, extended object tracking is performed to compare the recursive state estimation quality of various state-of-the-art LRKFs.

### 4.1. Moment Errors of a Standard Normal Distribution

First, we investigate how well the employed sampling schemes of state-of-the-art LRKFs approximate the moments of a standard normal distribution. Thus, we are interested in the expectation values

$$\mathbb{E}[x_1^{n_1} x_2^{n_2} \cdots x_N^{n_N}] = \int_{\mathbb{R}^N} x_1^{n_1} x_2^{n_2} \cdots x_N^{n_N} \mathcal{N}(\underline{x}; \underline{0}, \mathbf{I}_N) d\underline{x} ,$$

with

$$\sum_{i=1}^N n_i = m, \quad 0 \leq n_i \leq m$$

for different dimensions  $N$  and moment orders  $m$ . This has the advantage of being independent of a concrete system and measurement model. Note that for given  $N$  and  $m$ , there exists  $N^m$  possible combinations to select the values for  $n_i$ . Hence, a moment is characterized by  $N^m$  values.

As all state-of-the-art LRKFs employ a point symmetric sampling scheme and capture mean and covariance matrix, we focus on higher-order even moments. More precisely, we take a look at the 4th, 6th, and 8th moment, i.e.,  $m \in \{4, 6, 8\}$ . In many practical applications, 3D and 6D Gaussian distributions are of special interest. For example, the location and orientation in 2D or the position in 3D can be estimated using a three-dimensional system state. When additionally considering velocities in the 2D case or the orientation in the 3D case, a six-dimensional state is required. Thus, we chose to study the approximations of standard normal distributions with these two dimensions, i.e.,  $N \in \{3, 6\}$ .

To compare the different LRKF sampling techniques, for each dimension  $N$  and moment  $m$  we compute a normalized moment error according to

$$\sqrt{\frac{1}{N^m} \sum_{j=1}^{N^m} (\mathbb{E}_j^{\text{true}} - \mathbb{E}_j^{\text{LRKF}})^2},$$

where  $\mathbb{E}_j$  denotes one of the  $N^m$  possible combinations for the  $m$ th moment, the superscript "true" the true moment value and "LRKF" the LRKF sampling estimate. Note that the 8th moment of the 6D standard normal distribution is characterized by already over 1.5 million combinations.

We compare the new symmetric S<sup>2</sup>KF, the UKF with equally weighted samples, the RUKF, the fifth-degree CKF, and the GHKF with two quadrature points. As the symmetric S<sup>2</sup>KF and the RUKF do not have unique sample sets, we perform 100 Monte Carlo runs for both filters. In each run, for both filters new sample sets are generated, and the average moment error over all runs are computed. Moreover, the S<sup>2</sup>KF and the RUKF are evaluated with different number of samples. The results are depicted in Figures 3 and 4. As the UKF, the fifth-degree CKF, and the GHKF have a fixed number of samples, they are depicted as a bar at their respective employed number of samples.

The moment errors of the UKF and the S<sup>2</sup>KF are identical for the case when both filters use the same number of samples. This is due to the fact that both sample sets are equally weighted and the S<sup>2</sup>KF places its samples like the UKF (except for the rotation) as this minimize the utilized distance measure. The RUKF, however, scales the utilized UKF sample sets randomly. Consequently, its sample set is not necessarily equally weighted like for the UKF and the S<sup>2</sup>KF, and hence, their moment errors differ. Considering all moments, the S<sup>2</sup>KF delivers always smaller errors than the RUKF and for nearly all moments smaller errors than the GHKF (for the same number of samples). The sampling of the fifth-degree CKF is the only one that matches the 4th moment exactly. This is based on the fact that the spherical-radial rule of the fifth-degree CKF has a 5th-degree accuracy [12].

#### 4.2. Symmetric Measurement Equations

To illustrate the advantages of using a symmetric sampling scheme, we consider the two-dimensional system state

$$\underline{x} = [a, b]^\top$$

combined with the scalar and symmetric measurement equation

$$y = h(\underline{x}, v) = \sqrt{a^2 + b^2} + v,$$

where  $v$  is zero-mean Gaussian noise with variance  $\sigma^2 = 0.01$ . Hence, we measure a noisy distance from the system state  $\underline{x}$  to the state space origin. Such a symmetric measurement equation arises for example in [27, 28].

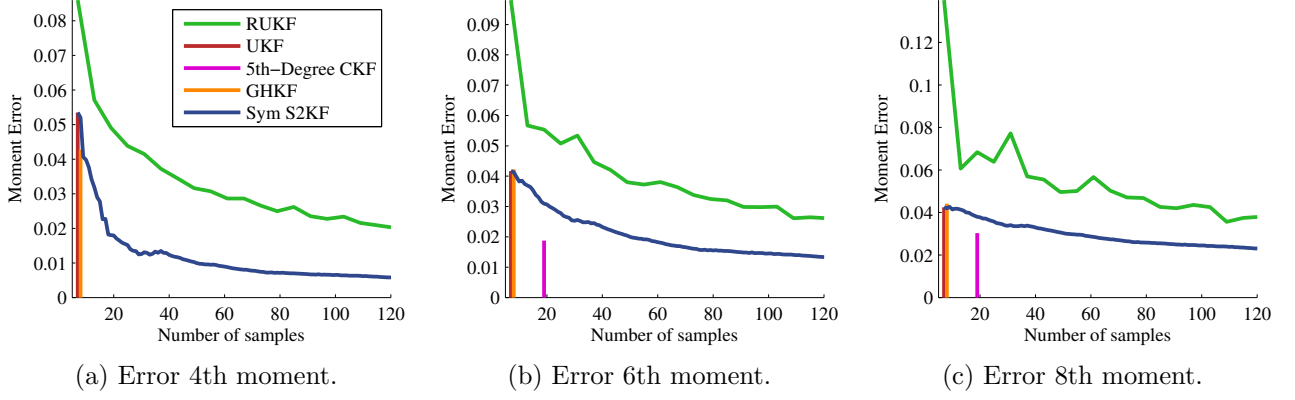


Figure 3: Moment errors of a 3D standard normal distribution.

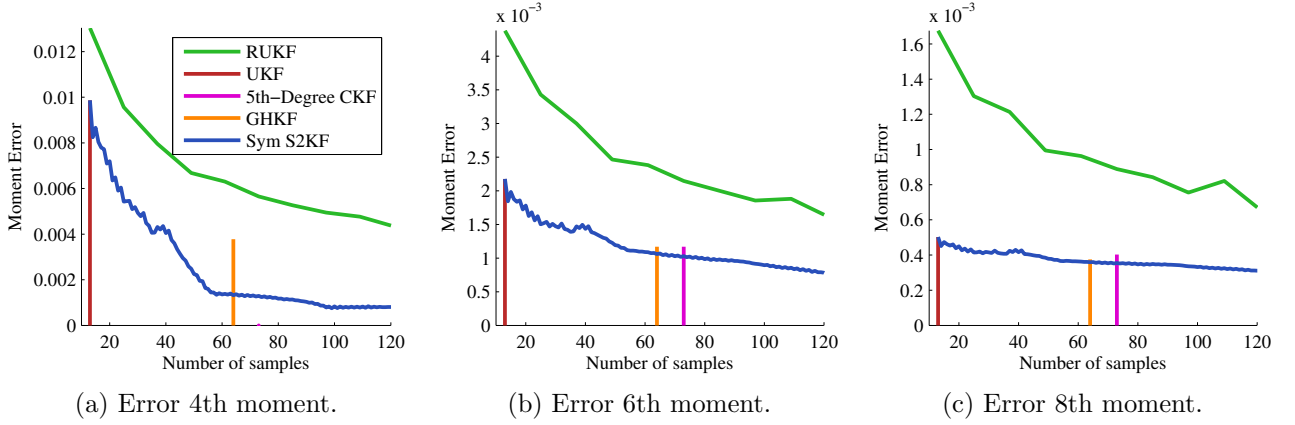


Figure 4: Moment errors of a 6D standard normal distribution.

We assume that the true system state is

$$\underline{x}_{true} = [1, 2]^\top ,$$

and our goal is to estimate it using a Nonlinear Kalman Filter initialized with mean and covariance matrix

$$\hat{\underline{x}}^p = [0, 0]^\top, \mathbf{C}^p = \text{diag}(4, 0.5) .$$

The setup is illustrated in Fig. 5. From the the estimator's perspective, the received measurement  $\tilde{y}$  could stem from any state located on the gray circle around the prior mean, not only  $\underline{x}_{true}$ . Hence, a Nonlinear Kalman Filter cannot gain any new information about the hidden system state from the measurement  $\tilde{y}$ . This situation is reflected in a zero cross-covariance matrix of state and measurement  $\mathbf{C}^{x,y}$  in (8) and (9). Consequently, the posterior state estimate (mean and covariance matrix) equals the prior, no matter what prior uncertainty we have.

Now, we try to reproduce this result when using LRKFs. More precisely, we compare the asymmetric S<sup>2</sup>KF, its new symmetric version (both using 11 samples), and the UKF. We perform  $R = 100$  Monte Carlo runs. In each run, we reset the initial state estimate, and simulate a noisy measurement  $\tilde{y}$  to perform one measurement update. Moreover, both S<sup>2</sup>KF variants compute a new set of samples approximating a standard normal distribution in every Monte Carlo run. We compute the Root Mean Square Error (RMSE) for the posterior mean

$$\sqrt{\frac{1}{R} \sum_{r=1}^R \|\hat{\underline{x}}_r^e - \hat{\underline{x}}^p\|_2^2} ,$$

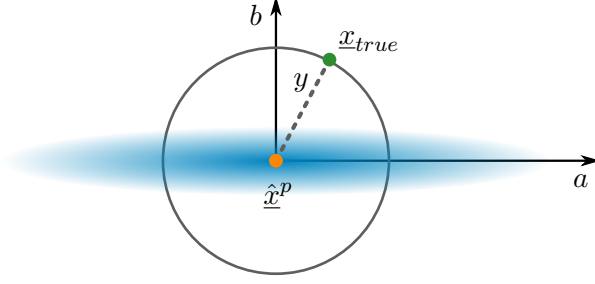


Figure 5: Symmetric measurement model with prior mean (orange), prior uncertainty (blue), true system state (green).

where  $\hat{\underline{x}}_r^e$  denotes the estimated posterior mean of run  $r$ . Additionally, we compute the RMSE of the posterior covariance matrix

$$\sqrt{\frac{1}{R} \sum_{r=1}^R \|\mathbf{C}_r^e - \mathbf{C}^p\|^2} ,$$

where  $\mathbf{C}_r^e$  denotes estimated posterior covariance matrix of run  $r$  and  $\|\cdot\|$  the Frobenius norm.

The results of the evaluation are depicted in Fig. 6. It can be seen that the asymmetric S<sup>2</sup>KF is the only filter with a small error. This can be explained with its asymmetric sampling scheme, as it makes a system state on the circle around the prior mean more likely than others. Hence, it introduces (theoretically non-existent) correlations between the measurement and the system state, i.e., a non-zero cross-covariance matrix. As a consequence, the asymmetric S<sup>2</sup>KF updates its state estimate mistakenly (albeit not that much). The other estimators do not have such a problem due to their symmetric sampling. So even such a simple scenario demonstrates the advantages of the new point symmetric sampling scheme of the S<sup>2</sup>KF.

#### 4.3. Extended Object Tracking

Now, we consider estimating the pose and extent of a cylinder in 3D based on a Random Hypersurface Model (RHM) [29, 30]. The system state is composed of position  $\underline{c}_k = [c_k^x, c_k^y, c_k^z]^\top$  and velocity  $\underline{\nu}_k = [\nu_k^x, \nu_k^y, \nu_k^z]^\top$ , rotation angles  $\underline{\phi}_k = [\phi_k^x, \phi_k^y]^\top$  and their velocities  $\underline{\omega}_k = [\omega_k^x, \omega_k^y]^\top$ , as well as the cylinder radius  $r_k$  and length  $l_k$  according to

$$\underline{x}_k = [\underline{c}_k^\top, \underline{\nu}_k^\top, \underline{\phi}_k^\top, \underline{\omega}_k^\top, r_k, l_k]^\top .$$

The temporal evolution of the cylinder is modeled with a constant velocity model

$$\underline{x}_k = \mathbf{A} \underline{x}_{k-1} + \underline{w} ,$$

with system matrix

$$\mathbf{A} = \begin{bmatrix} \mathbf{I}_3 & \mathbf{I}_3 & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_3 & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I}_2 & \mathbf{I}_2 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I}_2 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I}_2 \end{bmatrix}$$

and zero-mean Gaussian white noise  $\underline{w}$  with covariance matrix

$$\mathbf{C}^w = \text{diag}(10^{-6} \mathbf{I}_3, 10^{-4} \mathbf{I}_3, 10^{-10} \mathbf{I}_2, 10^{-5} \mathbf{I}_2, 10^{-4} \mathbf{I}_2) .$$

This linear model allows to compute the prediction step analytically for all LRKFs.

A measurement is a noisy point

$$\underline{\tilde{y}}_k = [\tilde{y}_k^x, \tilde{y}_k^y, \tilde{y}_k^z]^\top$$



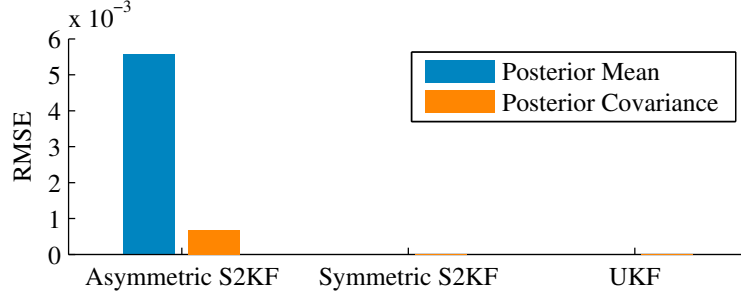


Figure 6: Estimation errors for symmetric measurement model.

from the cylinder's surface. It is related to the system state by means of the implicit nonlinear measurement equation

$$\underline{0} = h(\underline{x}_k, \tilde{\underline{y}}_k, \underline{v}, s) = \begin{bmatrix} (m_k^x)^2 + (m_k^y)^2 - r_k^2 \\ m_k^z - s \cdot l_k \\ (m_k^z - s \cdot l_k)^2 \end{bmatrix}, \quad (18)$$

where

$$\underline{m}_k = (\mathbf{R}(\phi_k^y) \cdot \mathbf{R}(\phi_k^x))^{-1}(\tilde{\underline{y}}_k - \underline{v} - \underline{c}_k),$$

and zero-mean Gaussian white noise  $\underline{v}$  with covariance matrix  $\mathbf{C}^v = 0.01 \cdot \mathbf{I}_3$  and multiplicative white noise  $s \sim \mathcal{U}(-0.5, 0.5)$ <sup>2</sup>. Furthermore,  $\mathbf{R}(\cdot)$  denotes a 3D rotation matrix around the respective axis. It is important to note that the measurement equation itself depends on the received measurement  $\tilde{\underline{y}}_k$ , and the estimator only takes the so-called pseudo measurement  $\underline{0}$  as input. The reason for this is that the proposed measurement model tries to minimize the Euclidean distance between the received measurements  $\tilde{\underline{y}}_k$  and the cylinder's surface, and thus, generates measurements of value zero in the optimal case. Note also that the quadratic term in the last row of (18) is necessary when dealing with multiplicative noise in combination with Kalman Filters [31, 29].

At each time step, we receive a set of 20 measurements

$$\mathcal{Y}_k = \{\tilde{\underline{y}}_k^{(1)}, \dots, \tilde{\underline{y}}_k^{(20)}\}.$$

As the order of processing measurements affects the filtered state estimate, we do not process measurements sequentially. More precisely, we process all measurements *at once*, that is, in a single measurement update, by stacking the measurements into a large measurement vector according to

$$\begin{bmatrix} \underline{0} \\ \vdots \\ \underline{0} \end{bmatrix} = \begin{bmatrix} h(\underline{x}_k, \tilde{\underline{y}}_k^{(1)}, \underline{v}^{(1)}, s^{(1)}) \\ \vdots \\ h(\underline{x}_k, \tilde{\underline{y}}_k^{(20)}, \underline{v}^{(20)}, s^{(20)}) \end{bmatrix}.$$

This, in turn, requires a set of  $20 \cdot 4 = 80$  measurement noise variables in total. Together with the twelve-dimensional system state, a LRKF has to sample a 92-dimensional random vector to perform a measurement update. The number of samples used by the investigated LRKFs are summarized in Table 1. It should be noted that the GHKF [11] is intractable for the considered scenario as it relies on a Cartesian product and would require at least  $2^{92}$  samples.

We simulate a nonlinear trajectory of a cylinder over 500 time steps including rotations in all its three degrees of freedom as depicted in Fig. 7. Additionally, the initial cylinder's length of 1 increases to 1.5 after 200 time steps, and the initial radius of 0.3 increases to 0.4 after further 100 time steps.

<sup>2</sup>As LRKFs can only sample Gaussian distributions, the uniform distribution will be approximated as a Gaussian using moment matching.

LRKF	Number of samples		
Fifth-degree CKF	$2 \cdot 92^2 + 1$	$=$	16,929
RUKF (with 5 iterations)	$5 \cdot (2 \cdot 92) + 1$	$=$	921
RUKF (with 20 iterations)	$20 \cdot (2 \cdot 92) + 1$	$=$	1,841
Asymmetric S <sup>2</sup> KF	Freely selectable		461
Asymmetric S <sup>2</sup> KF	Freely selectable		1,841
Symmetric S <sup>2</sup> KF	Freely selectable		461
Symmetric S <sup>2</sup> KF	Freely selectable		1,841

Table 1: LRKF settings for the measurement update.

Finally, at time step 400, the cylinder’s length shrinks back to 0.5. We perform 100 Monte Carlo runs. In each run, we initialize the estimators with

$$\begin{aligned}\hat{\mathbf{x}}_0^e &= [\hat{\mathbf{c}}^\top, 0, \dots, 0, 1, 2]^\top \\ \mathbf{C}_0^e &= \text{diag}(\mathbf{C}^c, 10^{-3}\mathbf{I}_3, 10^{-7}\mathbf{I}_4, 10^{-2}\mathbf{I}_2) \ ,\end{aligned}$$

where  $\hat{\mathbf{c}}$  denotes the mean and  $\mathbf{C}^c$  the covariance of the first set of measurements  $\mathcal{Y}_0$ . For each investigated LRKF, we compute the cylinder position RMSE (Fig. 8a), the RMSE of the angle between the true cylinder longitudinal axis and the estimated one (Fig. 8b), as well as the cylinder volume RMSE (Fig. 8c). Regarding the cylinder position, the RUKF instances were the filters with the largest errors although they used the same or twice the number of samples of the S<sup>2</sup>KF variants. The asymmetric S<sup>2</sup>KF was a little bit less accurate than the symmetric S<sup>2</sup>KF and the fifth-degree CKF. Same results can be observed for the cylinder orientation error. For the cylinder volume error, all estimators had noticeable error peaks at time steps 200, 300, and 400. These can be explained with the abrupt shape changes of the cylinder at the respective time steps. Furthermore, the fifth-degree CKF is not as good as in the other estimation quality criteria, and also the asymmetric S<sup>2</sup>KF is slightly better than the symmetric S<sup>2</sup>KF in the beginning.

However, when looking at the runtimes of the respective LRKF measurement updates in Fig. 8d, the fifth-degree CKF was the slowest filter due to its large amount of samples. The runtimes of the asymmetric and the symmetric S<sup>2</sup>KF were nearly identical as they used the same number of samples. For the case when the RUKF and the S<sup>2</sup>KF variants used the same number of samples, the RUKF was slower (11.5 ms compared to 4.5 ms) due to the additional overhead resulting from the creation of several 92-dimensional random orthogonal matrices during each measurement update. All in all, both S<sup>2</sup>KF variants were the filters yielding the best compromise between runtime performance and estimation accuracy. Moreover, this illustrates the advantage of being able to select the number of samples independently of the state/noise dimensions, in contrast to the fifth-degree CKF.

## 5. Conclusions

In this paper, we introduced a new point symmetric Gaussian sampling scheme for the Smart Sampling Kalman Filter. This reflects the point symmetry of the Gaussian distribution and allows for matching all odd moments of a standard normal distribution exactly. The new sampling technique does not only improve the estimation quality of the S<sup>2</sup>KF, it also speeds up the computation of the Gaussian samples as the number of Dirac mixture parameters to be optimized is reduced by half.

After describing the structure of a sample-based Kalman Filter, we extended the general Dirac mixture to a point symmetric form by distinguishing between an even and an odd number of samples. Then, we adapted the existing LCD distance measure to these new Dirac mixtures and also gave formulas for their respective gradients. These are required by the iterative optimization procedure which optimizes the Dirac mixture parameters to optimally approximate a multi-dimensional standard normal distribution with a set of equally weighted point symmetric samples. Furthermore, we improved

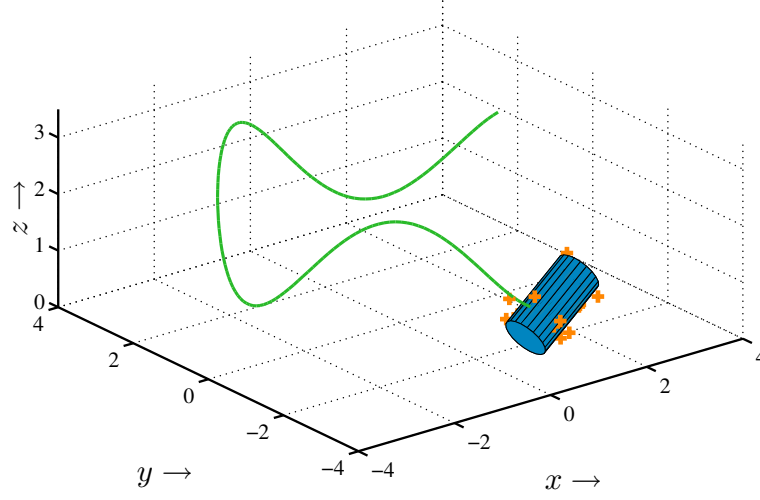
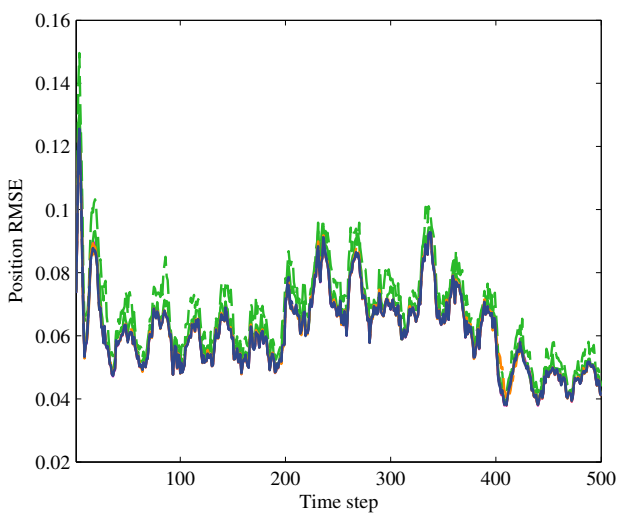
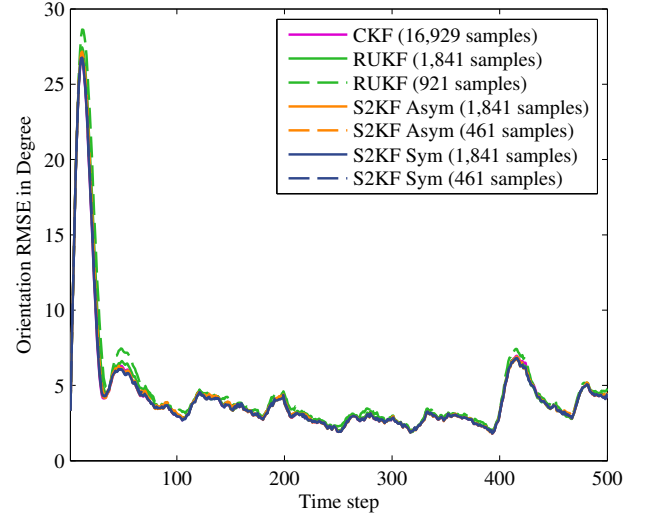


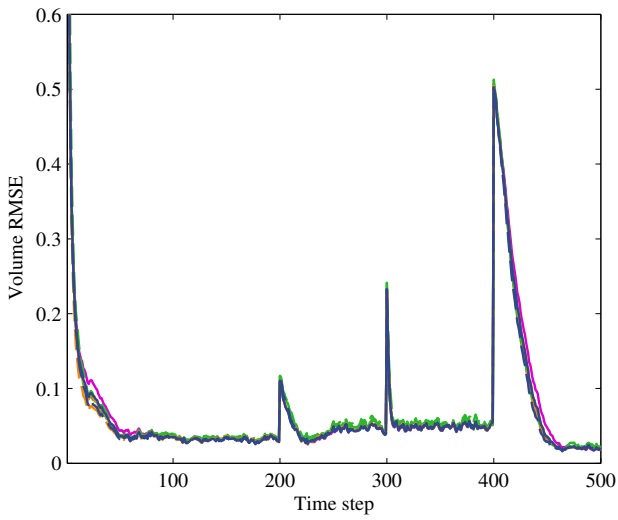
Figure 7: Cylinder state (blue) after 360 time steps inclusive its trajectory (green line) and 20 noisy measurements (orange crosses).



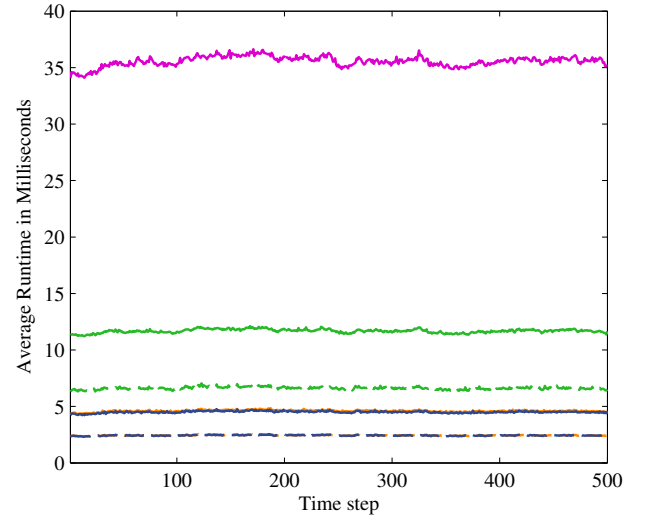
(a) Cylinder position error.



(b) Cylinder orientation error.



(c) Cylinder volume error.



(d) Measurement update runtime.

Figure 8: Cylinder tracking simulation results.

the numerical stability of the optimization, and together with the halved number of Dirac mixture parameters to be optimized, now it is possible to compute optimal approximations of a thousand-dimensional standard normal distribution comprising 10,000 samples. As the Progressive Gaussian Filter (PGF) [6] also relies on the S<sup>2</sup>KF Gaussian sampling, it can directly use and benefit from the new point symmetric sampling scheme.

The evaluations showed that the S<sup>2</sup>KF can handle symmetric measurement equations now much better when using the new symmetric sampling scheme. It was also shown that the S<sup>2</sup>KF gave the best compromise between estimation accuracy and filter runtime when dealing with high-dimensional problems such as extended object tracking. Additionally, this illustrated the advantage of the S<sup>2</sup>KF being able to use an arbitrary number of samples independent of the state/noise dimensions.

## Appendix A. Odd Moments of a Point Symmetric Dirac Mixture

The odd moments of an arbitrary density function  $f(\underline{x})$  with  $\underline{x} \in \mathbb{R}^N$  are defined as

$$\mathbb{E}_f[\prod_{j=1}^N x_j^{n_j}] = \int_{\mathbb{R}^N} \prod_{j=1}^N x_j^{n_j} \cdot f(\underline{x}) \, d\underline{x} \, ,$$

where

$$\sum_{j=1}^N n_j = 2k + 1, \quad 0 \leq n_j \leq 2k + 1, \quad k \in \mathbb{N} \, .$$

For a standard normal distribution, i.e.,  $f(\underline{x}) = \mathcal{N}(\underline{x}; \underline{0}, \mathbf{I}_N)$ , all odd moments equals zero. Hence, we have to show that this also holds for a point symmetric Dirac mixture density function comprising  $2L$  samples. By replacing the density  $f(\underline{x})$  with a point symmetric Dirac mixture approximation we obtain

$$\begin{aligned} \mathbb{E}_\delta[\prod_{j=1}^N x_j^{n_j}] &= \int_{\mathbb{R}^N} \prod_{j=1}^N x_j^{n_j} \frac{1}{2L} \sum_{i=1}^L \delta(\underline{x} - \underline{x}_i) + \delta(\underline{x} + \underline{x}_i) \, d\underline{x} \\ &= \frac{1}{2L} \sum_{i=1}^L \left( \prod_{j=1}^N x_{i,j}^{n_j} + \prod_{j=1}^N (-x_{i,j})^{n_j} \right) \\ &= \frac{1}{2L} \sum_{i=1}^L \left( \prod_{j=1}^N x_{i,j}^{n_j} - \prod_{j=1}^N x_{i,j}^{n_j} \right) = 0 \, . \end{aligned}$$

The same result can be easily obtained for the case of an odd number of samples  $2L + 1$  where the additional sample is placed at the state space origin.

## Appendix B. Proof of Distance $D^e(S)$

By using the facts that the distance  $D^e(S)$  is composed of sums of products of unnormalized Gaussians and their product is also an unnormalized Gaussian as well as the integral over a Gaussian equals always one, the three terms of the distance  $D^e(S)$  are obtained according to

$$D_1^e = \int_0^{b_{\max}} \frac{1}{\pi^{\frac{N}{2}} b^{N-1}} \int_{\mathbb{R}^N} \left( \frac{b^2}{1+b^2} \right)^N \cdot (2\pi)^N (1+b^2)^N \mathcal{N}(\underline{m}; \underline{0}, (1+b^2)\mathbf{I}_N)^2 \, d\underline{m} \, db$$

$$\begin{aligned}
&= \int_0^{b_{\max}} \frac{1}{\pi^{\frac{N}{2}} b^{N-1}} \left( \frac{b^2}{1+b^2} \right)^N \pi^{\frac{N}{2}} (1+b^2)^{\frac{N}{2}} db \\
&= \int_0^{b_{\max}} b \left( \frac{b^2}{1+b^2} \right)^{\frac{N}{2}} db ,
\end{aligned}$$

$$\begin{aligned}
D_2^e(S) &= \int_0^{b_{\max}} \frac{1}{\pi^{\frac{N}{2}} b^{N-1}} \int_{\mathbb{R}^N} \left( \frac{b^2}{1+b^2} \right)^{\frac{N}{2}} (2\pi)^{\frac{N}{2}} (1+b^2)^{\frac{N}{2}} \cdot \mathcal{N}(\underline{m}; \underline{0}, (1+b^2)\mathbf{I}_N) \cdot \frac{(2\pi)^{\frac{N}{2}} b^N}{2L} \cdot \\
&\quad \sum_{i=1}^L \mathcal{N}(\underline{m}; \underline{s}_i, b^2\mathbf{I}_N) + \mathcal{N}(\underline{m}; -\underline{s}_i, b^2\mathbf{I}_N) d\underline{m} db \\
&= \int_0^{b_{\max}} \frac{2^N \pi^{\frac{N}{2}} b^{N+1}}{2L(2\pi)^{\frac{N}{2}} (1+2b^2)^{\frac{N}{2}}} \cdot \sum_{i=1}^L \exp\left(-\frac{1}{2} \frac{\|\underline{s}_i\|_2^2}{(1+2b^2)}\right) + \exp\left(-\frac{1}{2} \frac{\|-\underline{s}_i\|_2^2}{(1+2b^2)}\right) db \\
&= \int_0^{b_{\max}} \frac{2b}{2L} \left( \frac{2b^2}{1+2b^2} \right)^{\frac{N}{2}} \cdot \sum_{i=1}^L \exp\left(-\frac{1}{2} \frac{\|\underline{s}_i\|_2^2}{(1+2b^2)}\right) db ,
\end{aligned}$$

and

$$\begin{aligned}
D_3^e(S) &= \int_0^{b_{\max}} \frac{1}{\pi^{\frac{N}{2}} b^{N-1}} \int_{\mathbb{R}^N} \left( \frac{(2\pi)^{\frac{N}{2}} b^N}{2L} \right)^2 \cdot \sum_{i=1}^L \mathcal{N}(\underline{m}; \underline{s}_i, b^2\mathbf{I}_N) + \mathcal{N}(\underline{m}; -\underline{s}_i, b^2\mathbf{I}_N) \cdot \\
&\quad \sum_{j=1}^L \mathcal{N}(\underline{m}; \underline{s}_j, b^2\mathbf{I}_N) + \mathcal{N}(\underline{m}; -\underline{s}_j, b^2\mathbf{I}_N) d\underline{m} db \\
&= \int_0^{b_{\max}} \frac{2^N \pi^{\frac{N}{2}} b^{N+1}}{(2L)^2 (2\pi)^{\frac{N}{2}} (2b^2)^{\frac{N}{2}}} \sum_{i=1}^L \sum_{j=1}^L \exp\left(-\frac{1}{2} \frac{\|\underline{s}_i - \underline{s}_j\|_2^2}{2b^2}\right) + \exp\left(-\frac{1}{2} \frac{\|\underline{s}_i + \underline{s}_j\|_2^2}{2b^2}\right) + \\
&\quad \exp\left(-\frac{1}{2} \frac{\|-\underline{s}_i - \underline{s}_j\|_2^2}{2b^2}\right) + \exp\left(-\frac{1}{2} \frac{\|\underline{s}_j - \underline{s}_i\|_2^2}{2b^2}\right) db \\
&= \int_0^{b_{\max}} \frac{2b}{(2L)^2} \sum_{i=1}^L \sum_{j=1}^L \exp\left(-\frac{1}{2} \frac{\|\underline{s}_i - \underline{s}_j\|_2^2}{2b^2}\right) + \exp\left(-\frac{1}{2} \frac{\|\underline{s}_i + \underline{s}_j\|_2^2}{2b^2}\right) db .
\end{aligned}$$

### Appendix C. Proof of Theorem 3.1

Like in [20], to compute the term  $D_3^e(S)$  we use that for  $z > 0$

$$\int_0^{b_{\max}} \frac{2}{b} \exp\left(-\frac{1}{2} \frac{z}{2b^2}\right) db = -\text{Ei}\left(-\frac{1}{2} \frac{z}{2b_{\max}^2}\right) , \tag{C.1}$$

where  $\text{Ei}(x)$  is the exponential integral defined as

$$\text{Ei}(x) = \int_{-\infty}^x \frac{e^t}{t} dt .$$

Moreover, the product rule gives

$$\frac{b_{\max}^2}{2} \exp\left(-\frac{1}{2} \frac{z}{2b_{\max}^2}\right) = \int_0^{b_{\max}} b \exp\left(-\frac{1}{2} \frac{z}{2b^2}\right) db + \frac{z}{4} \int_0^{b_{\max}} \frac{1}{b} \exp\left(-\frac{1}{2} \frac{z}{2b^2}\right) db ,$$

and together with (C.1) we obtain

$$\int_0^{b_{\max}} b \exp\left(-\frac{1}{2} \frac{z}{2b^2}\right) db = \frac{b_{\max}^2}{2} \exp\left(-\frac{1}{2} \frac{z}{2b_{\max}^2}\right) + \frac{z}{8} \text{Ei}\left(-\frac{1}{2} \frac{z}{2b_{\max}^2}\right). \quad (\text{C.2})$$

This directly results in the expression

$$D_3^e(S) = \frac{2}{(2L)^2} \sum_{i=1}^L \sum_{j=1}^L \frac{b_{\max}^2}{2} \left( \exp\left(-\frac{1}{2} \frac{\|\underline{s}_i - \underline{s}_j\|_2^2}{2b_{\max}^2}\right) + \exp\left(-\frac{1}{2} \frac{\|\underline{s}_i + \underline{s}_j\|_2^2}{2b_{\max}^2}\right) \right) + \frac{1}{8} \left( \|\underline{s}_i - \underline{s}_j\|_2^2 \text{Ei}\left(-\frac{1}{2} \frac{\|\underline{s}_i - \underline{s}_j\|_2^2}{2b_{\max}^2}\right) + \|\underline{s}_i + \underline{s}_j\|_2^2 \text{Ei}\left(-\frac{1}{2} \frac{\|\underline{s}_i + \underline{s}_j\|_2^2}{2b_{\max}^2}\right) \right).$$

#### Appendix D. Proof of Distance $D^o(S)$

The distance  $D^o(S)$  differs from its even counterpart due to the additional sample placed fixed at the state space origin. This does not effect  $D_1^o$ , and hence, it equals  $D_1^e$ . The other two terms are sums of their reweighted even counterparts (due to the changed sample weight) and terms comprising also products of unnormalized Gaussians. Hence, they are given as

$$\begin{aligned} D_2^o(S) &= \frac{2L}{2L+1} D_2^e(S) + \int_0^{b_{\max}} \frac{1}{\pi^{\frac{N}{2}} b^{N-1}} \int_{\mathbb{R}^N} \left( \frac{b^2}{1+b^2} \right)^{\frac{N}{2}} \cdot \\ &\quad (2\pi)^{\frac{N}{2}} (1+b^2)^{\frac{N}{2}} \mathcal{N}(\underline{m}; \underline{0}, (1+b^2)\mathbf{I}_N) \cdot \frac{(2\pi)^{\frac{N}{2}} b^N}{2L+1} \mathcal{N}(\underline{m}; \underline{0}, b^2\mathbf{I}_N) d\underline{m} db \\ &= \frac{2L}{2L+1} D_2^e(S) + \int_0^{b_{\max}} \frac{2^N \pi^{\frac{N}{2}} b^{N+1}}{2L+1} \frac{1}{(2\pi)^{\frac{N}{2}} (1+2b^2)^{\frac{N}{2}}} db \\ &= \frac{2L}{2L+1} D_2^e(S) + \int_0^{b_{\max}} \frac{b}{2L+1} \left( \frac{2b^2}{1+2b^2} \right)^{\frac{N}{2}} db \end{aligned}$$

and

$$\begin{aligned} D_3^o(S) &= \frac{(2L)^2}{(2L+1)^2} D_3^e(S) + \int_0^{b_{\max}} \frac{1}{\pi^{\frac{N}{2}} b^{N-1}} \cdot \int_{\mathbb{R}^N} \left( \frac{(2\pi)^{\frac{N}{2}} b^N}{2L+1} \right)^2 \cdot \\ &\quad \left( 2 \cdot \mathcal{N}(\underline{m}; \underline{0}, b^2\mathbf{I}_N) \cdot \left( \sum_{i=1}^L \mathcal{N}(\underline{m}; \underline{s}_i, b^2\mathbf{I}_N) + \mathcal{N}(\underline{m}; -\underline{s}_i, b^2\mathbf{I}_N) \right) + \mathcal{N}(\underline{m}; \underline{0}, b^2\mathbf{I}_N)^2 \right) d\underline{m} db \\ &= \frac{(2L)^2}{(2L+1)^2} D_3^e(S) + \int_0^{b_{\max}} \frac{2^N \pi^{\frac{N}{2}} b^{N+1}}{(2L+1)^2} \cdot \\ &\quad \frac{1}{(2\pi)^{\frac{N}{2}} (2b^2)^{\frac{N}{2}}} \left( 2 \cdot \sum_{i=1}^L \exp\left(-\frac{1}{2} \frac{\|\underline{s}_i\|_2^2}{2b^2}\right) + \exp\left(-\frac{1}{2} \frac{\|-\underline{s}_i\|_2^2}{2b^2}\right) + 1 \right) db \\ &= \frac{(2L)^2}{(2L+1)^2} D_3^e(S) + \frac{b_{\max}^2}{2(2L+1)^2} + \int_0^{b_{\max}} \frac{4b}{(2L+1)^2} \sum_{i=1}^L \exp\left(-\frac{1}{2} \frac{\|\underline{s}_i\|_2^2}{2b^2}\right) db. \end{aligned}$$

### Appendix E. Proof of Theorem 3.2

A closed-form expression for  $D_3^o(S)$  can directly be obtained by using again (C.2) as well as the closed-form expression for  $D_3^e(S)$  resulting in

$$D_3^o(S) = \frac{(2L)^2}{(2L+1)^2} D_3^e(S) + \frac{b_{\max}^2}{2(2L+1)^2} + \frac{4}{(2L+1)^2} \sum_{i=1}^L \frac{b_{\max}^2}{2} \exp\left(-\frac{1}{2} \frac{\|\underline{s}_i\|_2^2}{2b_{\max}^2}\right) + \frac{1}{8} \|\underline{s}_i\|_2^2 \text{Ei}\left(-\frac{1}{2} \frac{\|\underline{s}_i\|_2^2}{2b_{\max}^2}\right) .$$

### Appendix F. Boundedness of $D^e(S)$ and $D^o(S)$

We show the boundedness of the distances  $D^e(S)$  and  $D^o(S)$  for an increasing dimension  $N$ . For a given  $b_{\max}$  it holds

$$\begin{aligned} \lim_{N \rightarrow \infty} D_1^e &= \lim_{N \rightarrow \infty} \int_0^{b_{\max}} \underbrace{b \left( \frac{b^2}{1+b^2} \right)^{\frac{N}{2}}}_{\rightarrow 0 \text{ for } N \rightarrow \infty} db = 0 , \\ \lim_{N \rightarrow \infty} D_2^e(S) &= \lim_{N \rightarrow \infty} \int_0^{b_{\max}} \frac{2b}{2L} \left( \frac{2b^2}{1+2b^2} \right)^{\frac{N}{2}} \cdot \underbrace{\sum_{i=1}^L \exp\left(-\frac{1}{2} \frac{\|\underline{s}_i\|_2^2}{1+2b^2}\right)}_{\leq L} db \\ &\leq \lim_{N \rightarrow \infty} \int_0^{b_{\max}} \underbrace{b \left( \frac{2b^2}{1+2b^2} \right)^{\frac{N}{2}}}_{\rightarrow 0 \text{ for } N \rightarrow \infty} db = 0 , \end{aligned}$$

and

$$\begin{aligned} \lim_{N \rightarrow \infty} D_3^e(S) &= \lim_{N \rightarrow \infty} \int_0^{b_{\max}} \frac{2b}{(2L)^2} \cdot \underbrace{\sum_{i=1}^L \sum_{j=1}^L \exp\left(-\frac{1}{2} \frac{\|\underline{s}_i - \underline{s}_j\|_2^2}{2b^2}\right) + \exp\left(-\frac{1}{2} \frac{\|\underline{s}_i + \underline{s}_j\|_2^2}{2b^2}\right)}_{\leq 2L^2} db \\ &\leq \lim_{N \rightarrow \infty} \int_0^{b_{\max}} b db = \frac{b_{\max}^2}{2} . \end{aligned}$$

Hence, the distance  $D^e(S)$  is bounded by  $b_{\max}$  according to

$$\lim_{N \rightarrow \infty} D^e(S) = \lim_{N \rightarrow \infty} D_1^e - 2D_2^e(S) + D_3^e(S) \leq \frac{b_{\max}^2}{2} .$$

In a similar manner, the same result can be obtained for the distance  $D^o(S)$ .

### Appendix G. Proof of Theorem 3.3

A closed-form expression for  $\frac{\partial D_3^e(S)}{\partial s_i^{(d)}}$  can directly be obtained using (C.1) resulting in

$$\frac{\partial D_3^e(S)}{\partial s_i^{(d)}} = \frac{1}{(2L)^2} \sum_{j=1}^L (s_i^{(d)} - s_j^{(d)}) \text{Ei}\left(-\frac{1}{2} \frac{\|\underline{s}_i - \underline{s}_j\|_2^2}{2b_{\max}^2}\right) + (s_i^{(d)} + s_j^{(d)}) \text{Ei}\left(-\frac{1}{2} \frac{\|\underline{s}_i + \underline{s}_j\|_2^2}{2b_{\max}^2}\right) .$$

## Appendix H. Proof of Theorem 3.4

A closed-form expression for  $\frac{\partial D_3^o(S)}{\partial s_i^{(d)}}$  can directly be obtained using (C.1) and the closed-form expression for  $\frac{\partial D_3^s(S)}{\partial s_i^{(d)}}$  resulting in

$$\frac{\partial D_3^o(S)}{\partial s_i^{(d)}} = \frac{(2L)^2}{(2L+1)^2} \frac{\partial D_3^e(S)}{\partial s_i^{(d)}} + \frac{s_i^{(d)}}{(2L+1)^2} \text{Ei} \left( -\frac{1}{2} \frac{\|s_i\|_2^2}{2b_{\max}^2} \right).$$

## Appendix References

- [1] Rudolf E. Kalman, “A New Approach to Linear Filtering and Prediction Problems,” in *Transaction of the ASME - Journal of Basic Engineering*, Mar. 1960, pp. 35–45.
- [2] Sanjeev Arulampalam, Simon Maskell, Neil Gordon, and Tim Clapp, “A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking,” *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 174–188, Feb. 2002.
- [3] Branko Ristic, Sanjeev Arulampalam, and Neil Gordon, *Beyond the Kalman Filter: Particle Filters for Tracking Applications*. Artech House Publishers, 2004.
- [4] Arnaud Doucet and Adam M. Johansen, “A Tutorial on Particle Filtering and Smoothing: Fifteen Years Later,” in *Oxford Handbook of Nonlinear Filtering*, 2011, pp. 656–704.
- [5] Jayesh H. Kotecha and Petar M. Djuric, “Gaussian Particle Filtering,” *IEEE Transactions on Signal Processing*, vol. 51, no. 10, pp. 2592–2601, Oct. 2003.
- [6] Jannik Steinbring and Uwe D. Hanebeck, “Progressive Gaussian Filtering Using Explicit Likelihoods,” in *Proceedings of the 17th International Conference on Information Fusion (Fusion 2014)*, Salamanca, Spain, Jul. 2014.
- [7] Dan Simon, *Optimal State Estimation*, 1st ed. Wiley & Sons, 2006.
- [8] Tine Lefebvre, Herman Bruyninckx, and Joris De Schutter, “Kalman Filters for Non-Linear Systems: A Comparison of Performance,” *International Journal of Control*, vol. 77, no. 7, pp. 639–653, May 2004.
- [9] Simon J. Julier and Jeffrey K. Uhlmann, “Unscented Filtering and Nonlinear Estimation,” in *Proceedings of the IEEE*, vol. 92, Mar. 2004, pp. 401–422.
- [10] Ondřej Straka, Jindřich Duník, and Miroslav Šimandl, “Unscented Kalman Filter with Advanced Adaptation of Scaling Parameter,” *Automatica*, vol. 50, no. 10, pp. 2657–2664, Oct. 2014.
- [11] Kazufumi Ito and Kaiqi Xiong, “Gaussian Filters for Nonlinear Filtering Problems,” *IEEE Transactions on Automatic Control*, vol. 45, no. 5, pp. 910–927, May 2000.
- [12] Bin Jia, Ming Xin, and Yang Cheng, “High-Degree Cubature Kalman Filter,” *Automatica*, vol. 49, no. 2, pp. 510–518, Feb. 2013.
- [13] Jindřich Duník, Ondřej Straka, and Miroslav Šimandl, “The Development of a Randomised Unscented Kalman Filter,” in *Proceedings of the 18th IFAC World Congress*, Milano, Italy, Aug. 2011, pp. 8–13.
- [14] —, “Stochastic Integration Filter,” *IEEE Transactions on Automatic Control*, vol. 58, no. 6, pp. 1561–1566, Jun. 2013.
- [15] Ángel F. García-Fernández, Lennart Svensson, and Mark Morelande, “Iterated Statistical Linear Regression for Bayesian Updates,” in *Proceedings of the 17th International Conference on Information Fusion (Fusion 2014)*, Salamanca, Spain, Jul. 2014.
- [16] Pawe Stano, Zsófia Lendek, Jelmer Braaksma, Robert Babuska, Cees de Keizer, and Arnold J. den Dekker, “Parametric Bayesian Filters for Nonlinear Stochastic Dynamical Systems: A Survey,” *IEEE Transactions on Cybernetics*, vol. 43, no. 6, pp. 1607–1624, Dec. 2013.
- [17] Jannik Steinbring and Uwe D. Hanebeck, “LRKF Revisited: The Smart Sampling Kalman Filter (S<sup>2</sup>KF),” *Journal of Advances in Information Fusion*, vol. 9, no. 2, pp. 106–123, Dec. 2014.
- [18] —, “S<sup>2</sup>KF: The Smart Sampling Kalman Filter,” in *Proceedings of the 16th International Conference on Information Fusion (Fusion 2013)*, Istanbul, Turkey, Jul. 2013, pp. 2089–2096.
- [19] Cihan Ulas and Hakan Temeltas, “Planar Features and 6d-SLAM Based on Linear Regression Kalman Filters with N-Dimensional Approximated Gaussians,” in *Proceedings of the 19th IFAC World Congress*, Cape Town, South Africa, Aug. 2014, pp. 10 194–10 199.
- [20] Uwe D. Hanebeck, Marco F. Huber, and Vesa Klumpp, “Dirac Mixture Approximation of Multivariate Gaussian Densities,” in *Proceedings of the 2009 IEEE Conference on Decision and Control (CDC 2009)*, Shanghai, China, Dec. 2009.
- [21] Igor Gilitschenski and Uwe D. Hanebeck, “Efficient Deterministic Dirac Mixture Approximation of Gaussian Distributions,” in *Proceedings of the 2013 American Control Conference (ACC 2013)*, Washington D. C., USA, Jun. 2013.
- [22] Igor Gilitschenski, Jannik Steinbring, Uwe D. Hanebeck, and Miroslav Šimandl, “Deterministic Dirac Mixture Approximation of Gaussian Mixtures,” in *Proceedings of the 17th International Conference on Information Fusion (Fusion 2014)*, Salamanca, Spain, Jul. 2014.



- [23] Yaakov Bar-Shalom, X. Rong Li, and Thiagalingam Kirubarajan, *Estimation with Applications to Tracking and Navigation*. New York Chichester Weinheim Brisbane Singapore Toronto: Wiley-Interscience, 2001.
- [24] Wolfgang Härdle and Léopold Simar, *Applied Multivariate Statistical Analysis*, 2nd ed. Berlin Heidelberg: Springer, 2008.
- [25] Uwe D. Hanebeck and Vesa Klumpp, “Localized Cumulative Distributions and a Multivariate Generalization of the Cramér-von Mises Distance,” in *Proceedings of the 2008 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI 2008)*, Seoul, Republic of Korea, Aug. 2008, pp. 33–39.
- [26] Jorge Nocedal and Stephen J. Wright, *Numerical Optimization*, 2nd ed., ser. Springer Series in Operations Research and Financial Engineering. Springer, 2006.
- [27] Marcus Baum and Uwe D. Hanebeck, “The Kernel-SME Filter for Multiple Target Tracking,” in *Proceedings of the 16th International Conference on Information Fusion (Fusion 2013)*, Istanbul, Turkey, Jul. 2013.
- [28] Christof Chlebek and Uwe D. Hanebeck, “Bayesian Approach to Direct Pole Estimation,” in *Proceedings of the 2014 European Control Conference (ECC 2014)*, Strasbourg, France, Jun. 2014.
- [29] Florian Faion, Marcus Baum, and Uwe D. Hanebeck, “Tracking 3d Shapes in Noisy Point Clouds with Random Hypersurface Models,” in *Proceedings of the 15th International Conference on Information Fusion (Fusion 2012)*, Singapore, Jul. 2012.
- [30] Marcus Baum and Uwe D. Hanebeck, “Extended Object Tracking with Random Hypersurface Models,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 50, no. 1, pp. 149–159, Jan. 2014.
- [31] Marcus Baum, Florian Faion, and Uwe D. Hanebeck, “Modeling the Target Extent with Multiplicative Noise,” in *Proceedings of the 15th International Conference on Information Fusion (Fusion 2012)*, Singapore, Jul. 2012, pp. 2406–2412.